

TCC Ciência de Dados e Big Data

ANÁLISE DE PROGRAMAS DE
SERVIÇOS DE STREAMING

Introdução

Com o aumento dos serviços de streaming e a quantidade de produções sendo feitas, mesmo com as muitas reclamações dos clientes informando que muitas produções não são de boa qualidade, esse projeto se destina a tentar prever as pontuações com base em outra pontuação muito famosa a IMDB e classificar os filmes das streaming entre “Bom” e “Ruim”, baseado nos critérios do que eu considero bons.

Problema Proposto

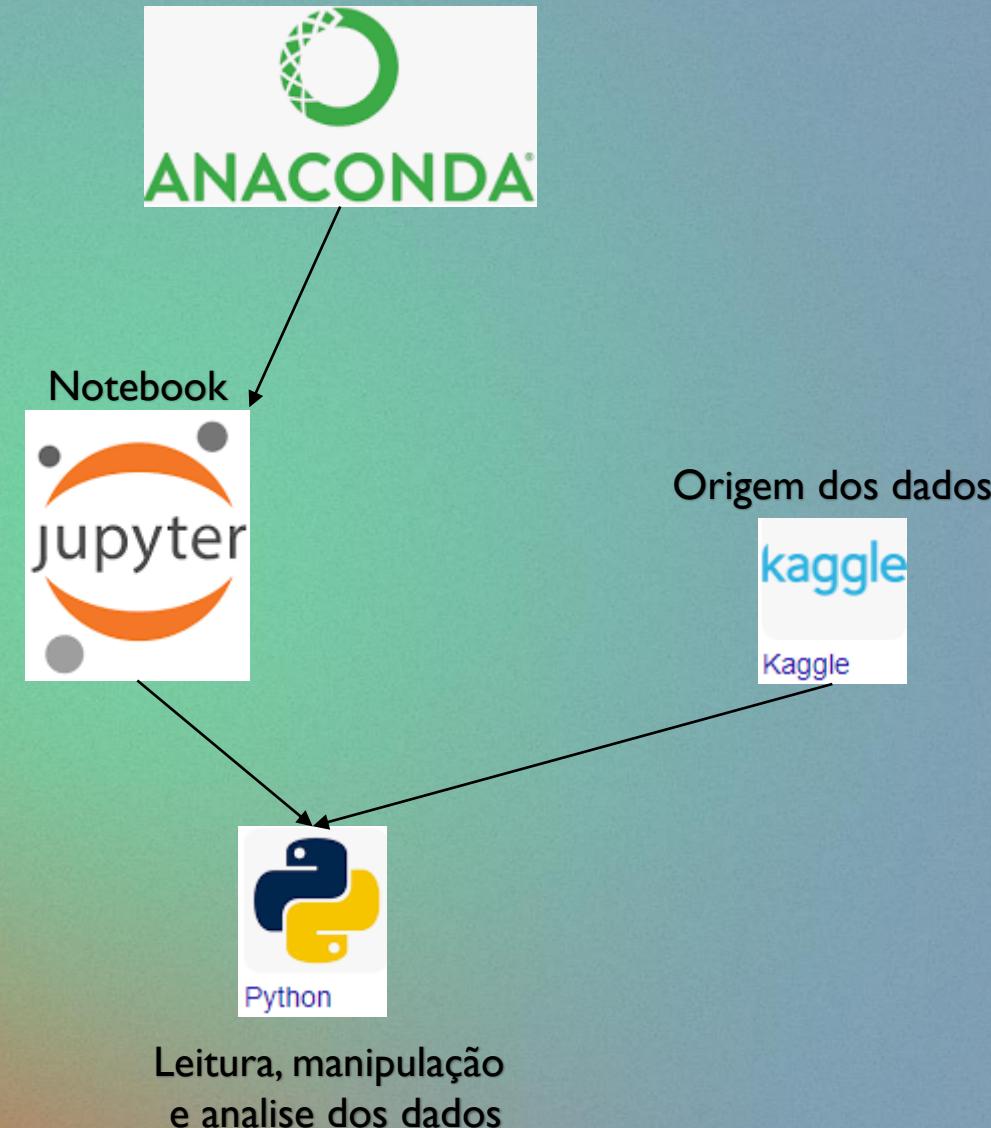
Criar um modelo para tentar prever a pontuação dos programas oferecidos pelas serviços de streaming e classificar esses programas entre os que são “Bom” e “Ruim” a partir de alguns critérios.

1. Pontuação do imdb igual ou maior que 7
2. Pontuação do tmdb igual ou superior a 6
3. Runtime igual ou superior a 70
4. Quantidade de votos do imdb igual ou superior 400

Ferramentas utilizadas

ETAPAS

- I. Coleta de dados
2. Processamento e tratamento dos dados
3. Análise e exploração dos dados
4. Criação dos modelos de Machine Learning



Coleta de dados

KAGGLE

Streaming	Produções	link
Amazon	Filmes e Series	https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows
Netflix	Filmes e Series	https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies
HBO	Filmes e Series	https://www.kaggle.com/datasets/victorsoeiro/hbo-max-tv-shows-and-movies

streaming	Amazon	Netflix
type		
MOVIE	2566	3196
SHOW	442	1823

Datasets

Bases usadas para treinar o modelo

		title	type	release_year	runtime	genres	production_countries	imdb_score	imdb_votes	tmdb_score	streaming
1	Taxi Driver	MOVIE		1976	114	['drama', 'crime']	['US']	8.2	808582.0	8.179	Netflix
2	Deliverance	MOVIE		1972	109	['drama', 'action', 'thriller', 'european']	['US']	7.7	107673.0	7.300	Netflix
3	Monty Python and the Holy Grail	MOVIE		1975	91	['fantasy', 'action', 'comedy']	['GB']	8.2	534486.0	7.811	Netflix
4	The Dirty Dozen	MOVIE		1967	150	['war', 'action']	['GB', 'US']	7.7	72662.0	7.600	Netflix
5	Monty Python's Flying Circus	SHOW		1969	30	['comedy', 'european']	['GB']	8.8	73424.0	8.306	Netflix
...
5838	Happiness Ever After	MOVIE		2021	99	['drama', 'romance']	['ZA']	4.2	163.0	7.300	Netflix
5842	Super Monsters: Once Upon a Rhyme	MOVIE		2021	25	['animation', 'family']	[]	5.6	38.0	6.300	Netflix
5843	My Bride	MOVIE		2021	93	['romance', 'comedy', 'drama']	['EG']	5.0	327.0	5.300	Netflix
5847	Lokillo	MOVIE		2021	90	['comedy']	['CO']	3.8	68.0	6.300	Netflix
5849	Mighty Little Bheem: Kite Festival	SHOW		2021	7	['family', 'animation', 'comedy']	[]	7.8	18.0	10.000	Netflix

		title	type	release_year	runtime	genres	production_countries	imdb_score	imdb_votes	tmdb_score	streaming
0	The Three Stooges	SHOW		1934	19	['comedy', 'family', 'animation', 'action', 'f...	['US']	8.6	1092.0	7.6	Amazon
1	The General	MOVIE		1926	78	['action', 'drama', 'war', 'western', 'comedy']...	['US']	8.2	89766.0	8.0	Amazon
2	The Best Years of Our Lives	MOVIE		1946	171	['romance', 'war', 'drama']	['US']	8.1	63026.0	7.8	Amazon
3	His Girl Friday	MOVIE		1940	92	['comedy', 'drama', 'romance']	['US']	7.8	57835.0	7.4	Amazon
4	In a Lonely Place	MOVIE		1950	94	['thriller', 'drama', 'romance']	['US']	7.9	30924.0	7.6	Amazon
...
9843	Ammaa Ki Boli	MOVIE		2021	117	['comedy', 'drama']	['IN']	7.3	1335.0	1.0	Amazon
9844	Alleyway	MOVIE		2021	67	['action', 'crime', 'thriller']	[]	5.4	92.0	6.8	Amazon
9847	Girls' Night In	MOVIE		2021	91	['comedy', 'drama']	['US']	2.8	28.0	7.0	Amazon
9856	Anbirkiniyal	MOVIE		2021	118	['thriller', 'drama']	['IN']	6.8	361.0	7.0	Amazon
9864	Gun and a Hotel Bible	MOVIE		2021	58	['drama']	[]	4.0	142.0	6.5	Amazon

Base usada para testar o modelos

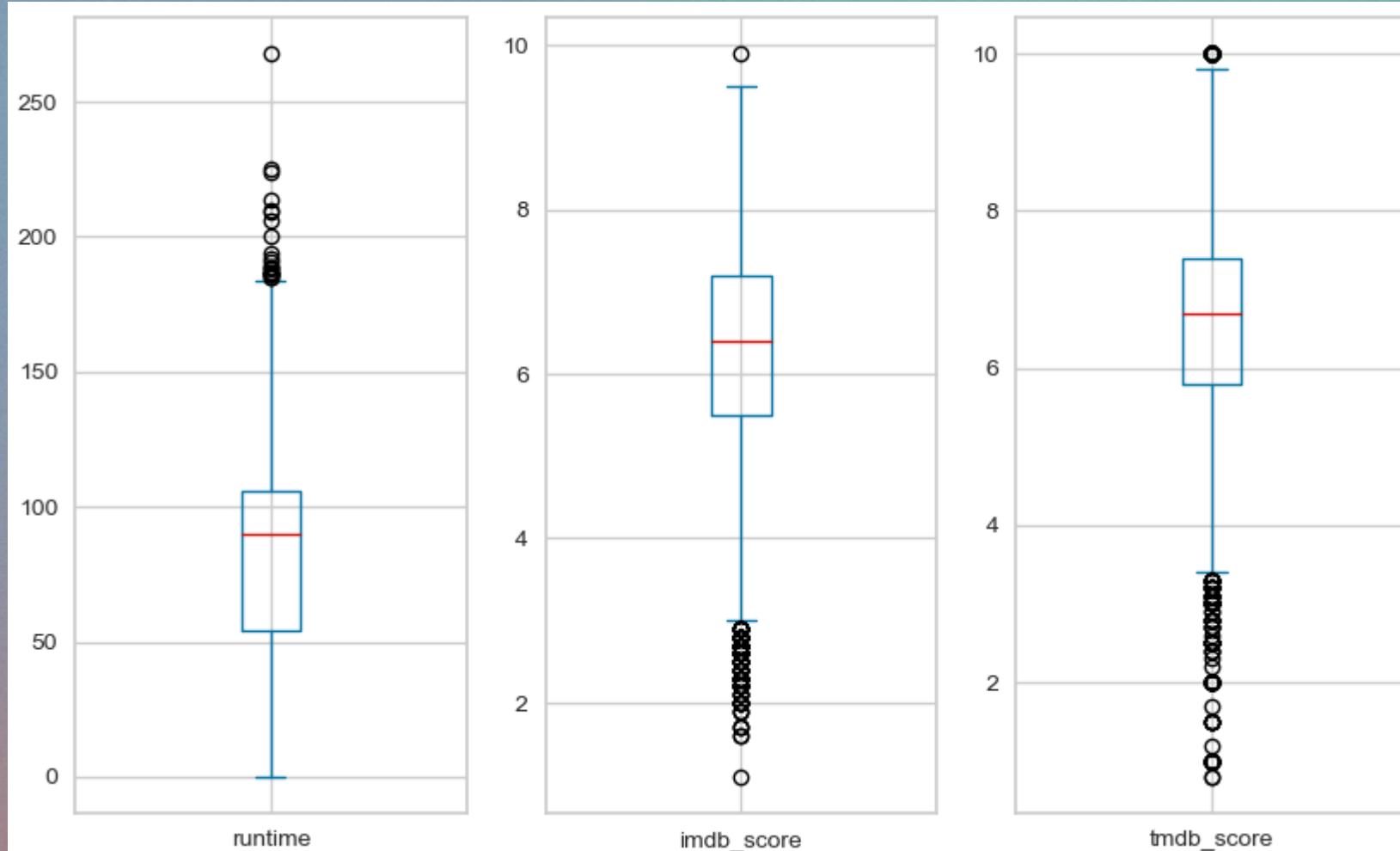
		title	type	release_year	runtime	genres	production_countries	imdb_score	imdb_votes	tmdb_score	streaming
0	The Wizard of Oz	MOVIE		1939	102	['fantasy', 'family']	['US']	8.1	389774.0	7.6	HBO
1	Citizen Kane	MOVIE		1941	119	['drama']	['US']	8.3	433804.0	8.0	HBO
2	Casablanca	MOVIE		1942	102	['drama', 'romance', 'war']	['US']	8.5	558849.0	8.2	HBO
3	The Big Sleep	MOVIE		1946	116	['thriller', 'crime']	['US']	7.9	84494.0	7.7	HBO
4	The Maltese Falcon	MOVIE		1941	100	['thriller', 'romance', 'crime']	['US']	8.0	156603.0	7.8	HBO
...
3275	Breathless	MOVIE		2021	106	['crime', 'drama', 'thriller']	['DO']	6.3	27.0	5.9	HBO
3279	Furry Friends Forever: Elmo Gets a Puppy	MOVIE		2021	26	['animation']	['US']	6.8	14.0	10.0	HBO
3283	Marlon Wayans: You Know What It Is	MOVIE		2021	58	['comedy']	['US']	3.8	224.0	5.4	HBO
3284	Ahir Shah: Dots	MOVIE		2021	61	['comedy']	[]	5.8	69.0	7.0	HBO
3290	Algo Azul	MOVIE		2021	90	['comedy']	['PA']	5.9	50.0	2.0	HBO

1. Dataset com dados Netflix

2. Dataset com dados Amazon

3. Dataset com dados HBO

Análise dos dados

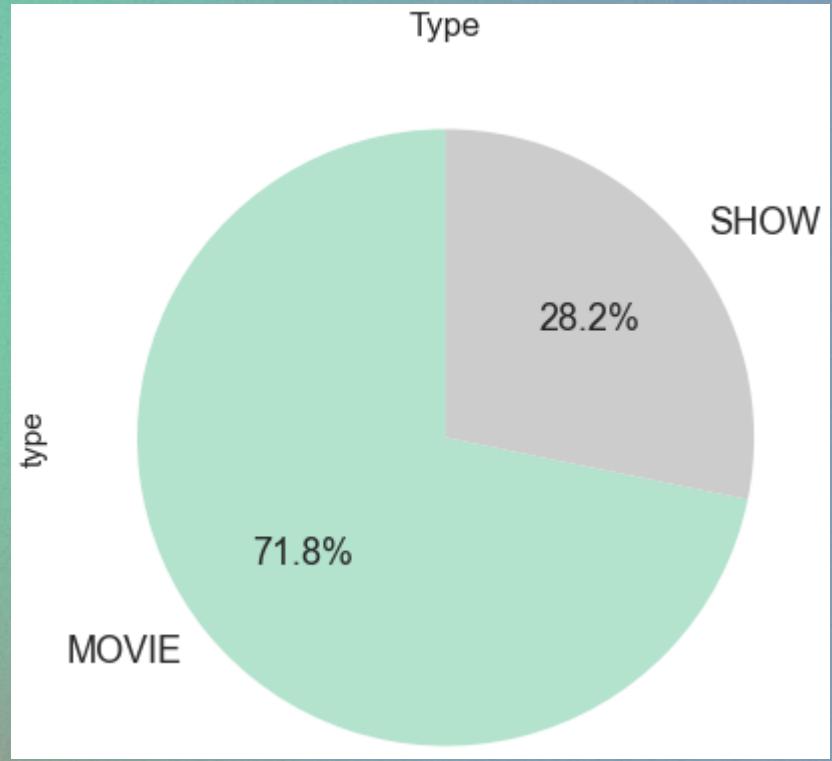
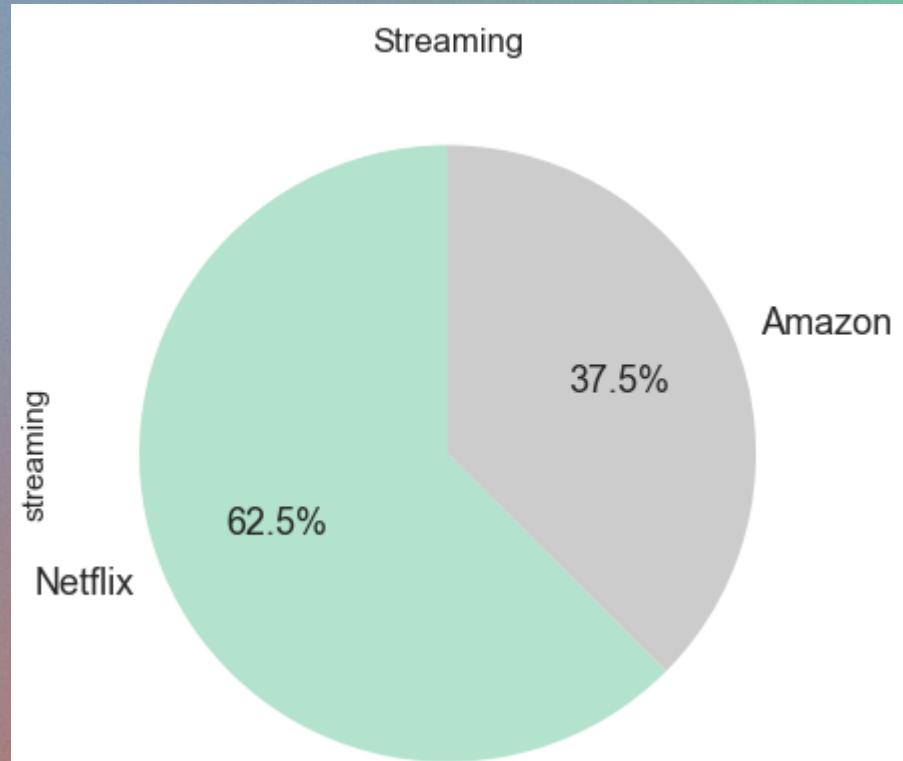


Presença de outliers
Runtime
Imdb_score
Tmdb_score

Não houve remoção
dos outliers e nem
nenhuma técnica de
dimensionalidade

Os dados não foram
facionados

Análise dos dados – Representatividades dos streamings e os tipos de produções



Modelos ML usados

Regressão Linear

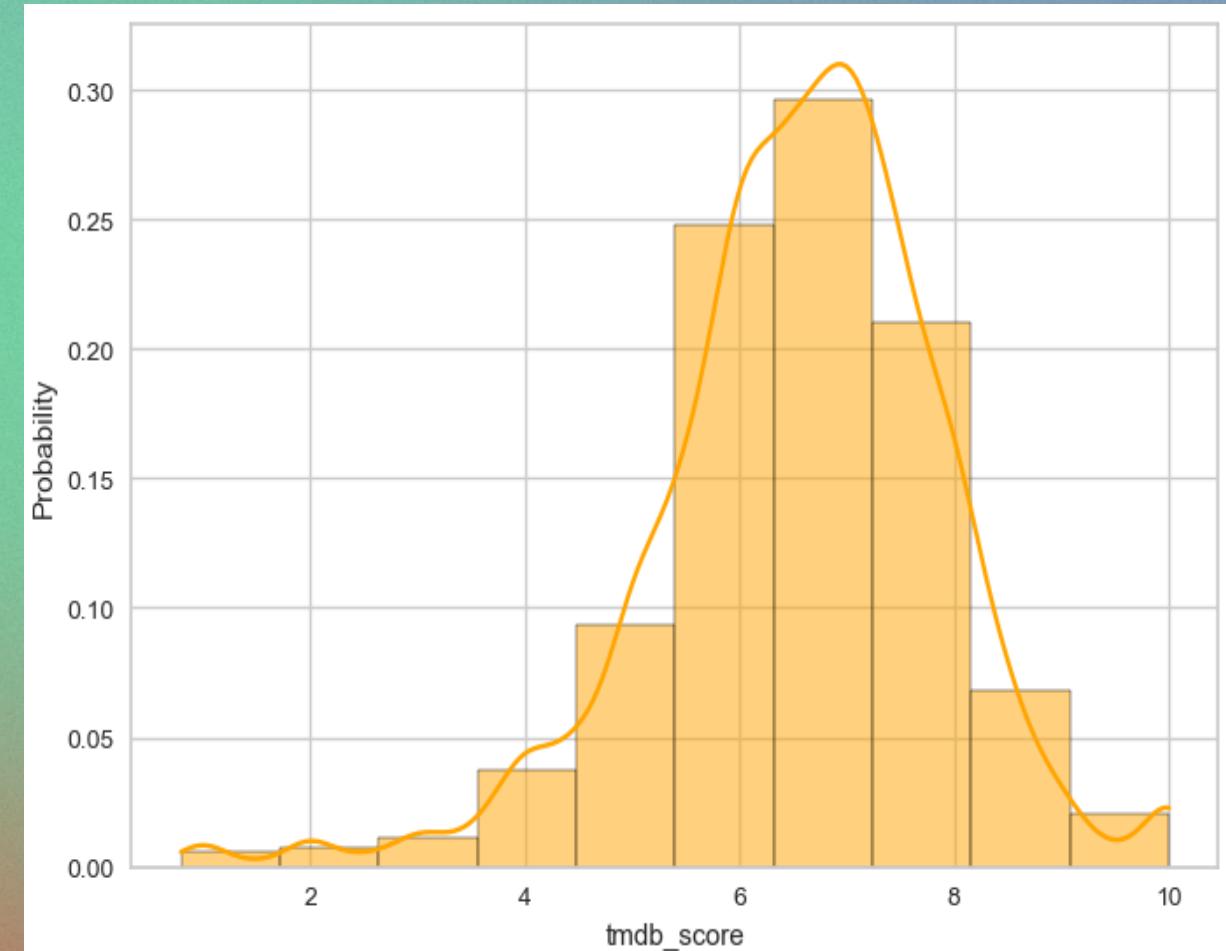
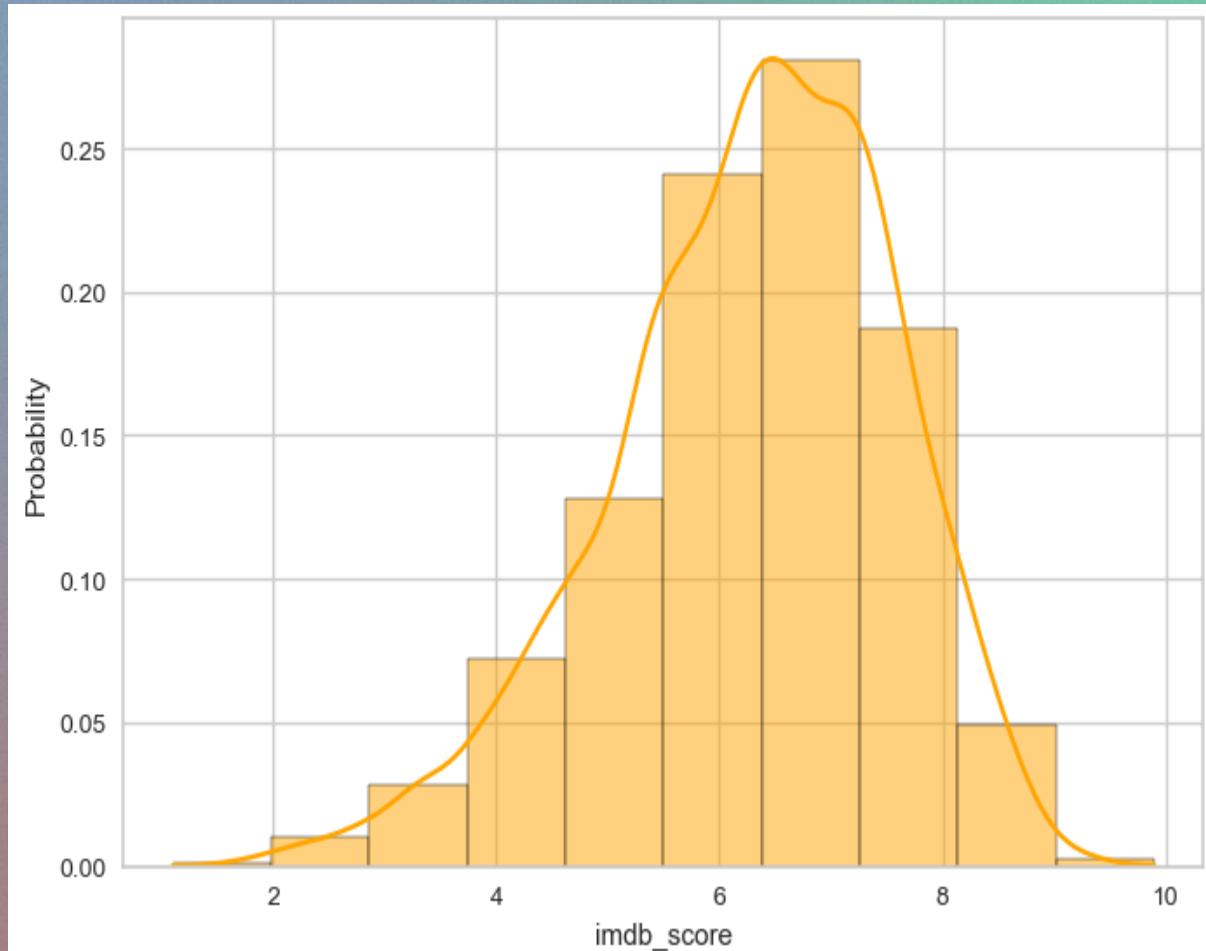
Baseado no aprendizado supervisionado, através da regressão.

Objetivo é descobrir a relação entre as 2 grandezas(IMDB e TMDB) para poder realizar previsões futuras

- Vantagens
 - I. Simples de implementar
 - 2. Prever o futuro
 - 3. Possível realizar ajustes ou pode se aplicar técnicas de dimensionalidade, técnicas de regularização e validação cruzada.
- Desvantagens
 - I. Os outliers podem impactar muito a previsão
 - 2. Limitado a relacionamentos lineares
 - 3. Os dados devem ser independentes

Regressão Linear– Histograma

Pelo gráfico do histograma entre as 2 grandes IMDB e TMDB, parecem ser comportar quase de forma identifica.



Regressão Linear– Correlação dos dados

Coeficiente de correlação (r)	Correlação Positiva	Coeficiente de correlação (r)	Correlação Negativa
$r = 1$	Perfeita	$r = -1$	Perfeita
$0,95 \leq r < 1$	Muito forte	$-0,95 \leq r < -1$	Muito forte
$0,8 \leq r < 0,95$	Forte	$-0,8 \leq r < -0,95$	Forte
$0,5 \leq r < 0,8$	Moderada	$-0,5 \leq r < -0,8$	Moderada
$0 \leq r < 0,5$	Fraca	$0 \leq r < -0,5$	Fraca

Correlação entre as 2 grandezas é de apenas 57%. É muito baixa

	imdb_score	tmdb_score
imdb_score	1.000000	0.502499
tmdb_score	0.502499	1.000000

Amazon

	release_year	runtime	imdb_score	imdb_votes	tmdb_score
release_year	1.000000	-0.049670	-0.019386	-0.100526	0.095120
runtime	-0.049670	1.000000	-0.160509	0.104245	-0.257727
imdb_score	-0.019386	-0.160509	1.000000	0.188369	0.568936
imdb_votes	-0.100526	0.104245	0.188369	1.000000	0.122148
tmdb_score	0.095120	-0.257727	0.568936	0.122148	1.000000

Dataset - Final

Analisando separadamente as 2 fontes, correlação se mantém baixa nos 2.

	imdb_score	tmdb_score
imdb_score	1.000000	0.580052
tmdb_score	0.580052	1.000000

Netflix

Regressão Linear – Representatividades dos streamings e os tipos de produções

OLS Regression Results						
Dep. Variable:	imdb_score	R-squared:	0.324			
Model:	OLS	Adj. R-squared:	0.324			
Method:	Least Squares	F-statistic:	3841.			
Date:	Thu, 02 Feb 2023	Prob (F-statistic):	0.00			
Time:	20:04:45	Log-Likelihood:	-11807.			
No. Observations:	8027	AIC:	2.362e+04			
Df Residuals:	8025	BIC:	2.363e+04			
Df Model:	1					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
Intercept	2.7694	0.058	47.817	0.000	2.656	2.883
tmdb_score	0.5364	0.009	61.974	0.000	0.519	0.553
Omnibus:	880.065	Durbin-Watson:	1.903			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1772.540			
Skew:	-0.703	Prob(JB):	0.00			
Kurtosis:	4.823	Cond. No.	33.7			

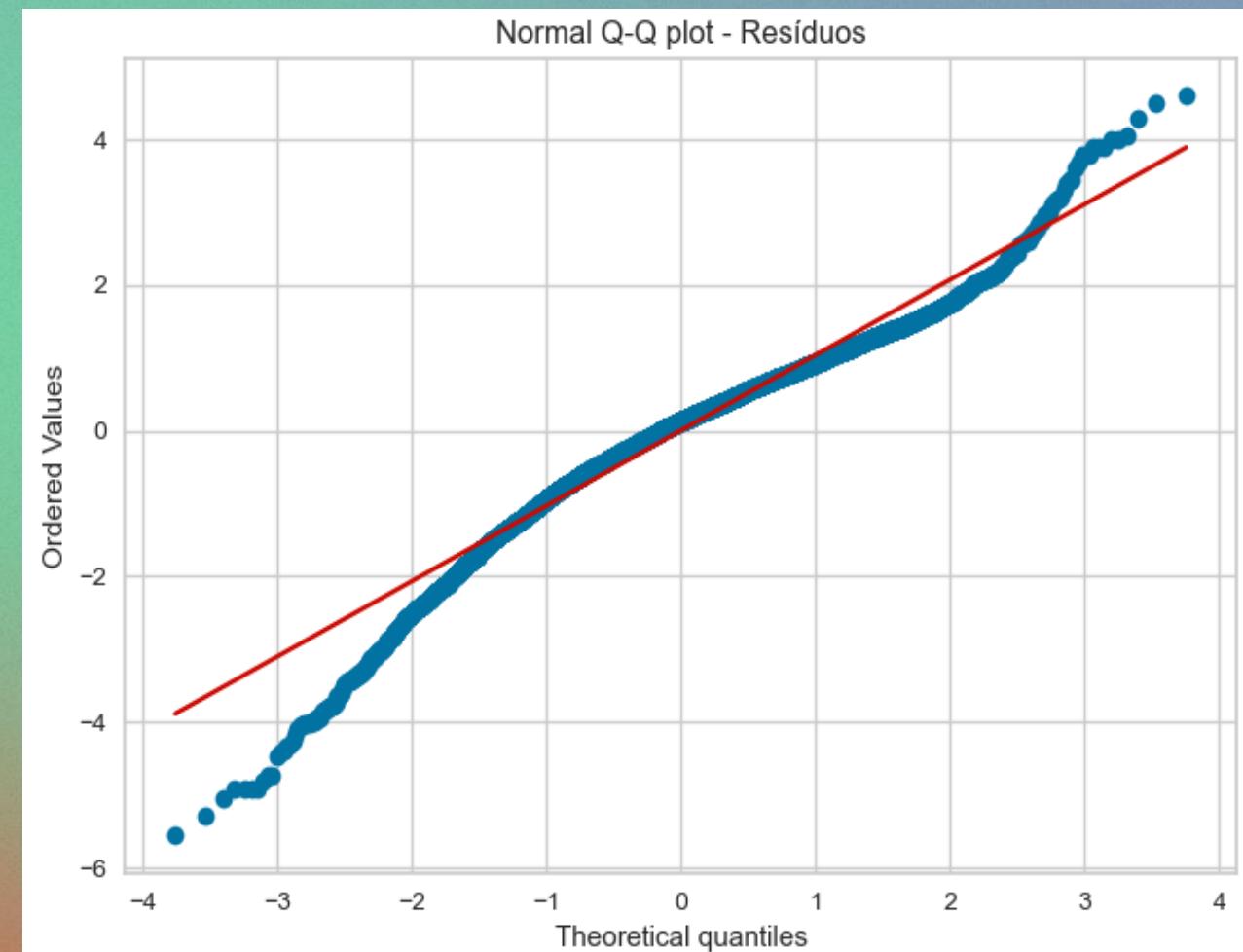
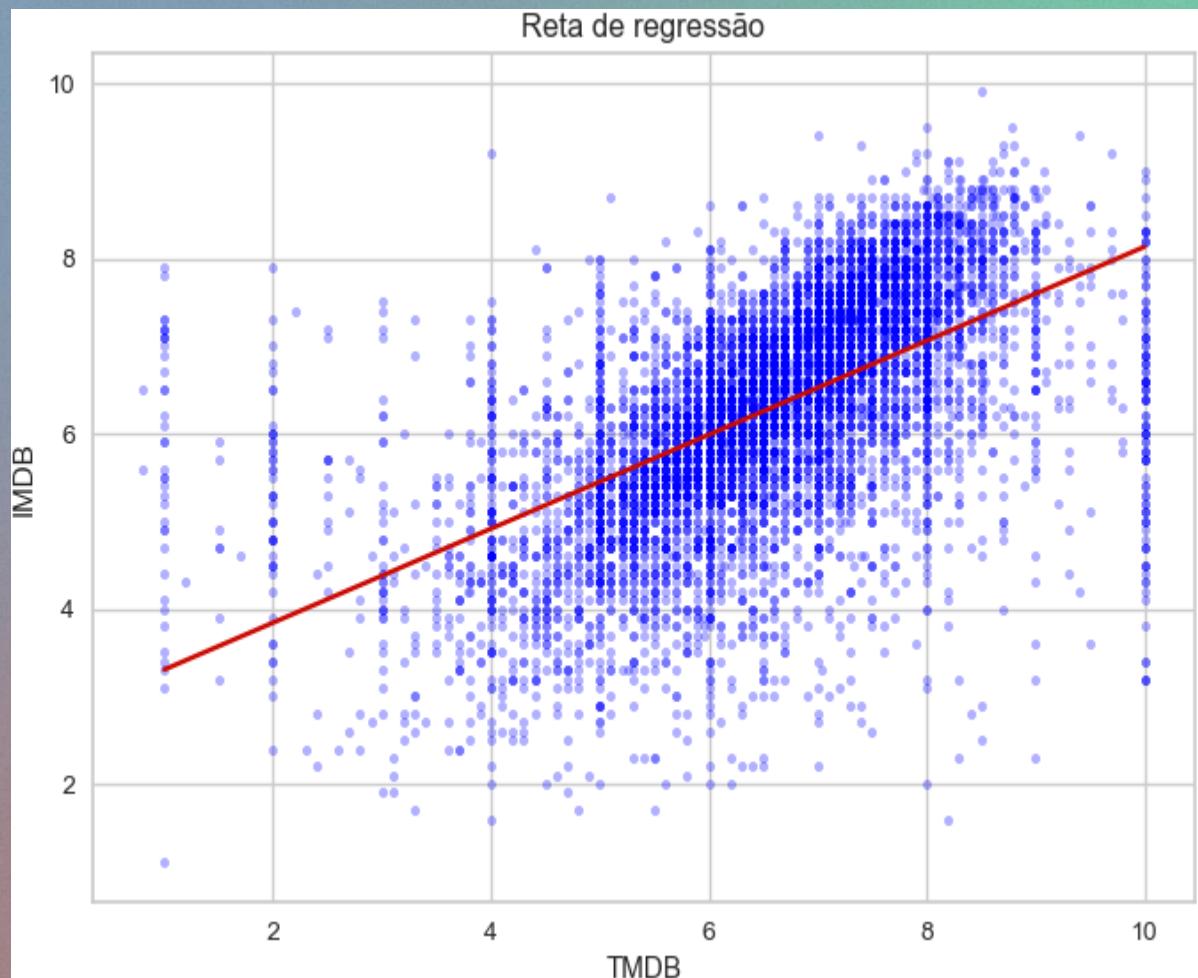
O resultado dos cálculos da regressão entre imdb e tmdb, deu um valor R-Quadrática Ajustada muito baixa, de apenas 32%

Coeficientes angular e linear de 0.54 e 2.77

Linear, corresponde o valor no ponto em que a reta corta o eixo das ordenadas
Angular, corresponde ao valor que dá a inclinação da reta.

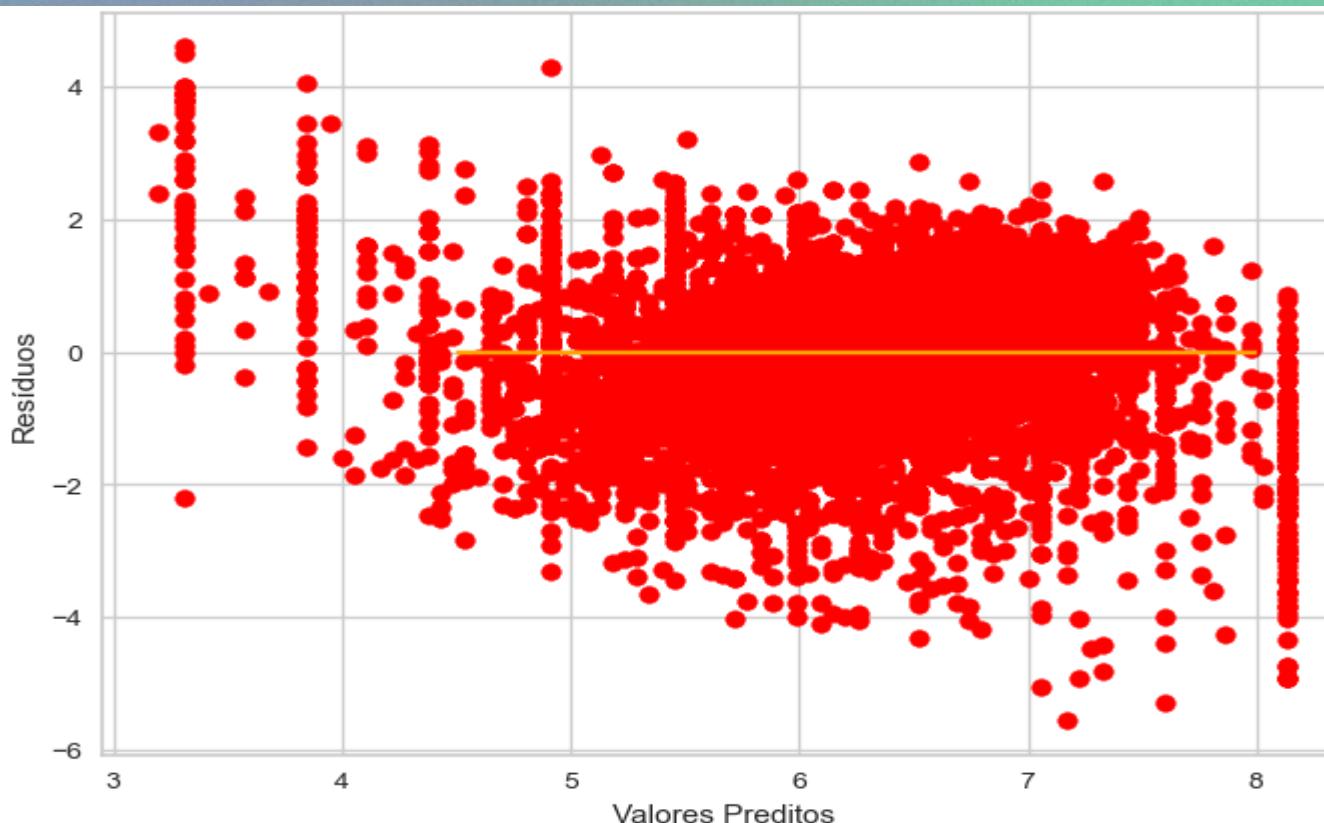
Regressão Linear– Reta de Regressão e Normal Q-Q

Aparentemente em algum momento se comporta como uma distribuição normal, com uma boa previsão no meio do eixo X, mas com grande margem de erro nas extremidades.



Regressão Linear– Testes estatísticos e Homocedasticidade

```
if ad_stat < ad_critico[2]:  
    print("Com " + str(100 - ad_teorico[2]) + "% de confiança, os dados são similares a uma distribuição normal"  
        +"segundo o teste de AD")  
else:  
    print("Com " + str(100 - ad_teorico[2]) + "% de confiança, os dados não são similares a uma distribuição normal"  
        +"segundo o teste de AD")  
  
Com 95.0% de confiança, os dados não são similares a uma distribuição normal segundo o teste de AD
```



Variação constante dos resíduos, tem que gerar algo parecido com um retângulo

Valor da estatística tem que ser menor que o crítico na 2 posição

```
statsmodels.stats.diagnostic.lilliefors(dataSet.imdb_score, dist="norm")  
  
(0.05887773223296189, 0.000999999999998899)  
O Valor da estatística calculada: 66.51603506568972  
Os valores críticos: [0.576 0.656 0.787 0.918 1.091]  
Os níveis de significância: [15. 10. 5. 2.5 1. ]
```

Teste Anderson

```
statsmodels.stats.diagnostic.lilliefors(dataSet.imdb_score, dist="norm")  
  
(0.05887773223296189, 0.000999999999998899)
```

Teste Lilliefors

P-Valor > 0,05

```
stats.shapiro(residuos)  
ShapiroResult(statistic=0.9657446146011353, pvalue=6.8007396681072725e-40)
```

Teste Shapiro-Wilk
P-Valor > 0,05

Modelos ML usados

Naive Bayes

Baseado na construção de classificadores, rótulos de classe representados como vetores.

Pressupõe que o valor de um recurso é independente do outro recurso, alterar o valor de um recurso não afetaria o valor do outro recurso.

Recomendado para grandes volumes de dados, previsão em tempo real (Mais rápido e está sempre pronto para aprender)

- Vantagens

1. Fácil de implementar.
2. Rápido
3. Se a suposição de independência se mantiver, ela funcionará com mais eficiência do que outros algoritmos.
4. Requer menos dados de treinamento.
5. É altamente escalável.
6. Pode fazer previsões probabilísticas.
7. Pode lidar com dados contínuos e discretos.
8. Insensível a características irrelevantes.
9. Pode funcionar facilmente com valores ausentes.
10. Fácil de atualizar na chegada de novos dados.
11. Mais adequado para problemas de classificação de texto.

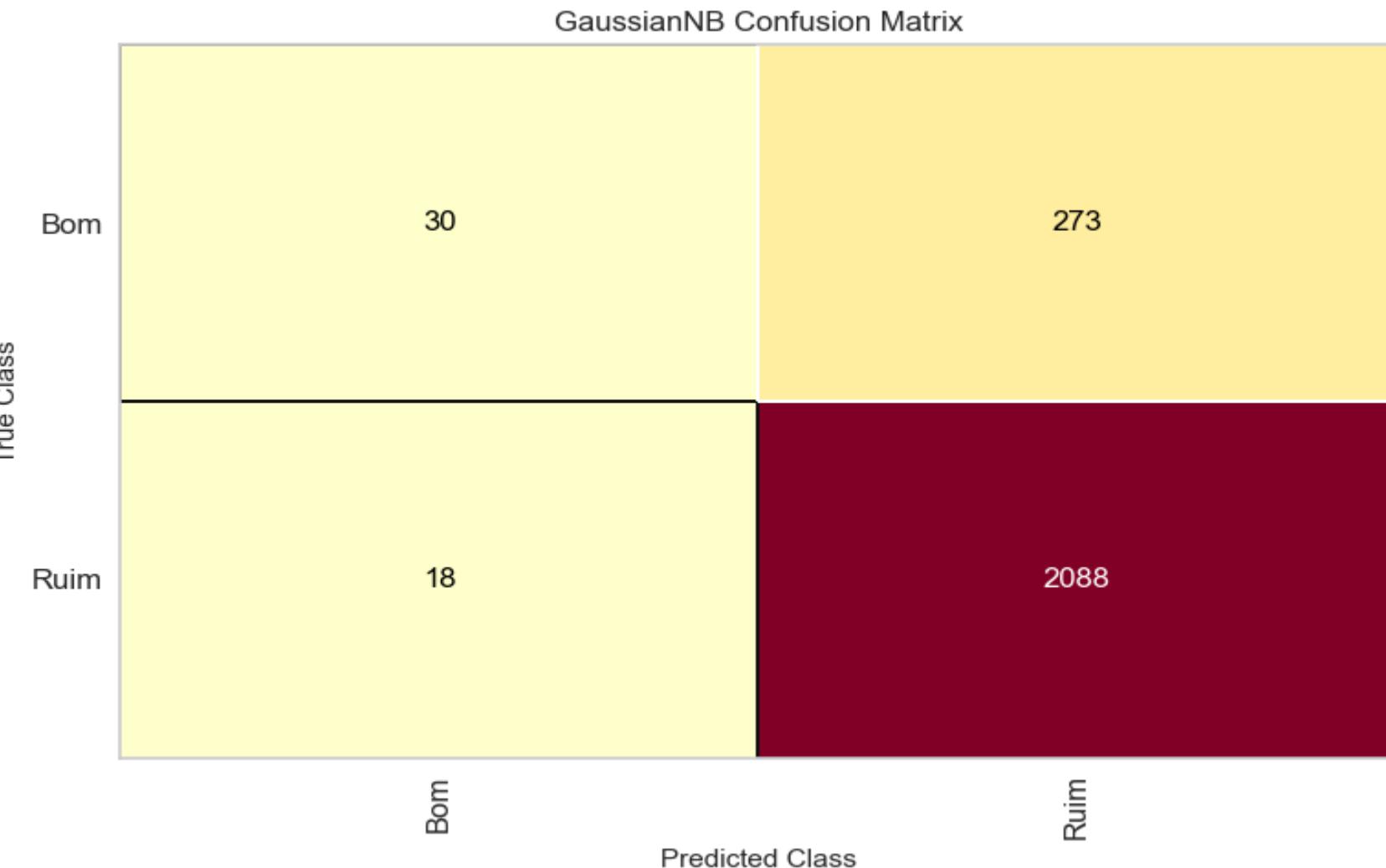
- Desvantagens

1. A forte suposição de que os recursos são independentes, o que dificilmente se aplica às aplicações da vida real.
2. Escassez de dados.
3. Chances de perda de precisão.
4. Frequência zero, isto é, se a categoria de qualquer variável categórica não for vista no conjunto de dados de treinamento, o modelo atribuirá uma probabilidade zero a essa categoria e, portanto, uma previsão não poderá ser feita.

Naive Bayes – Separação dos previsores da classificação

```
# Transformação dos atributos categóricos em atributos numéricicos, passando o índice de cada coluna categórica
labelencoder1 = LabelEncoder()
previsores[:,1] = labelencoder1.fit_transform(previsores[:,1])
```

```
# Separar as previsões da classificação (Status)
previsoes = dataSet.iloc[:,0:5].values
classe = dataSet.iloc[:,5].values
```



```
# Criação e treinamento do modelo
naive_bayes = GaussianNB()
naive_bayes.fit(x_treinamento, y_treinamento)
```

```
# Geração da matriz de confusão e cálculo da taxa de acerto e erro
confusao = confusion_matrix(y_teste, previsoes)
confusao
```

```
array([[ 30, 273],
       [ 18, 2088]], dtype=int64)
```

```
# Taxa de acerto
taxa_acerto = accuracy_score(y_teste, previsoes)
taxa_erro = 1 - taxa_acerto
taxa_acerto
```

```
0.8792029887920298
```

Taxa de acerto foi de 88%
Nos quadrantes BOM teve
30 acerto e no de RUIM
deve acerto de 2088

É um modelo válido para
ser usado, nessa análise.

Modelos ML usados

Árvore de decisão

Aprendizado de máquina supervisionado, que usa algoritmos de classificação e regressão.

Consiste de nós, nó-raiz e nós-folha.

Funciona de maneira recursiva, analisando cada novo nó.

Índice GINI :Com o cálculo do índice GINI, será verificada a distribuição dos dados nas variáveis preditoras de acordo com a variação da variável target, porém com um método diferente.A variável preditora com o menor índice Gini será a escolhida para o nó principal da árvore, pois um baixo valor do índice indica maior ordem na distribuição dos dados.

- Vantagens

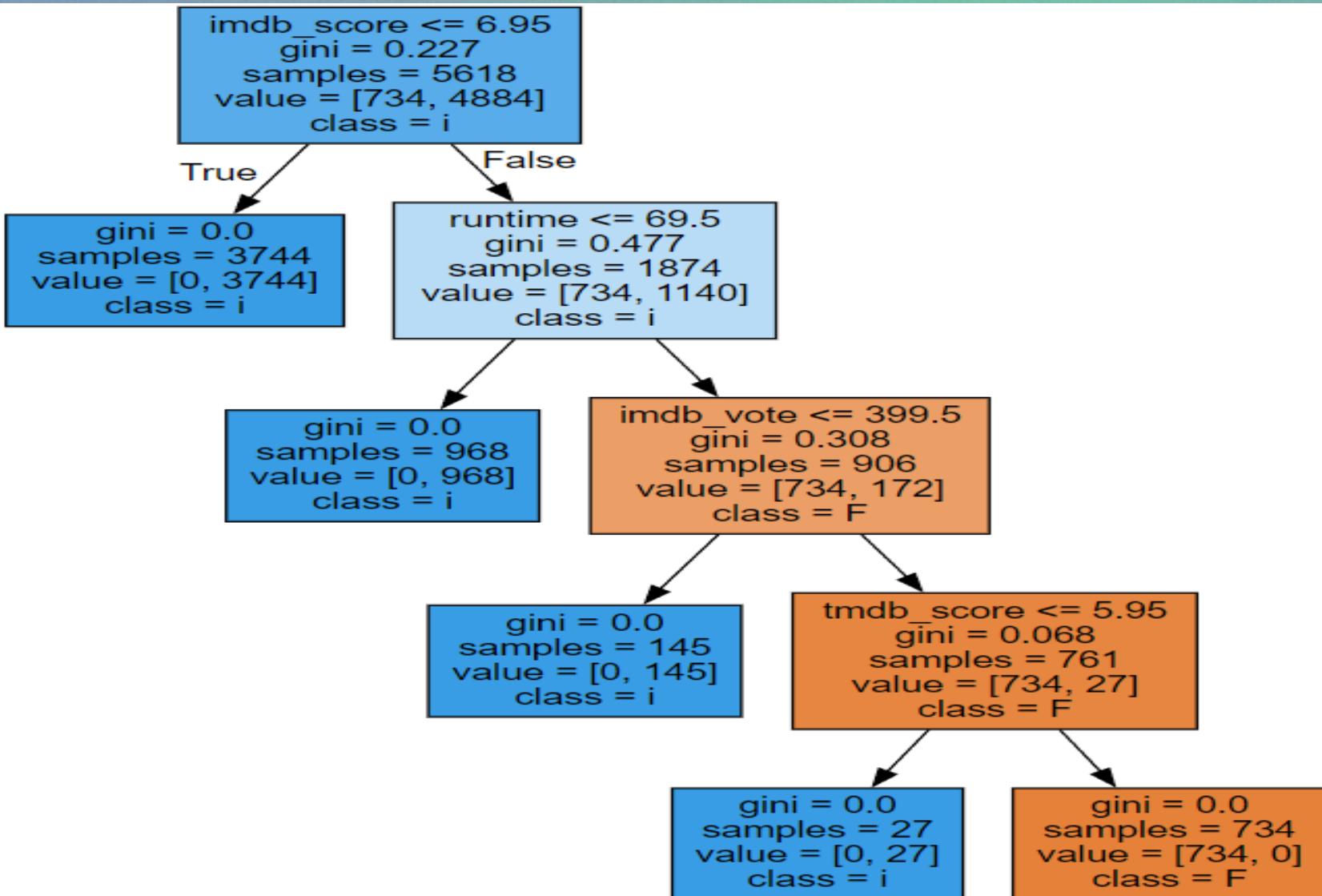
1. Fácil de entender:A visualização de uma árvore de decisão torna o problema fácil de compreender, mesmo para pessoas que não tenham perfil analítico. Não requer nenhum conhecimento estatístico para ler e interpretar. Sua representação gráfica é muito intuitiva e permite relacionar as hipóteses também facilmente.
2. Útil em exploração de dados:A árvore de decisão é uma das formas mais rápidas de identificar as variáveis mais significativas e a relação entre duas ou mais variáveis. Com a ajuda de árvores de decisão, podemos criar novas variáveis/características que tenham melhores condições de predizer a variável alvo.
3. Menor necessidade de limpar dados: Requer menos limpeza de dados em comparação com outras técnicas de modelagem. Até um certo nível, não é influenciado por pontos fora da curva “outliers” nem por valores faltantes (“missing values”).
4. Não é restrito por tipos de dados: Pode manipular variáveis numéricas e categóricas.
5. Método não paramétrico:A árvore de decisão é considerada um método não-paramétrico. Isto significa que as árvores de decisão não pressupõem a distribuição do espaço nem a estrutura do classificador.

- Desvantagens

1. Sobre ajuste (“Over fitting”): Sobre ajuste é uma das maiores dificuldades para os modelos de árvores de decisão. Este problema é resolvido através da definição de restrições sobre os parâmetros do modelo e da poda (discutido em mais detalhes abaixo).
2. Não adequado para variáveis contínuas: ao trabalhar com variáveis numéricas contínuas, a árvore de decisão perde informações quando categoriza variáveis em diferentes categorias.

Árvore de Decisão – Árvore gerada

```
# Transformação dos atributos categóricos em atributos numéricos, passando o índice de cada coluna categórica
labelencoder2 = LabelEncoder()
previsores[:,1] = labelencoder2.fit_transform(previsores[:,1])
```



```
# Separar as previsões da classificação (Status)
previsores = dataSet.iloc[:,0:5].values
classe = dataSet.iloc[:,5].values
```

```
#Criação e treinamento do modelo - Árvore de decisão
arvore = DecisionTreeClassifier(
    max_depth = None,
    max_features = None,
    min_samples_leaf=1,
    min_samples_split=2
)
arvore.fit(x_treinamento, y_treinamento)

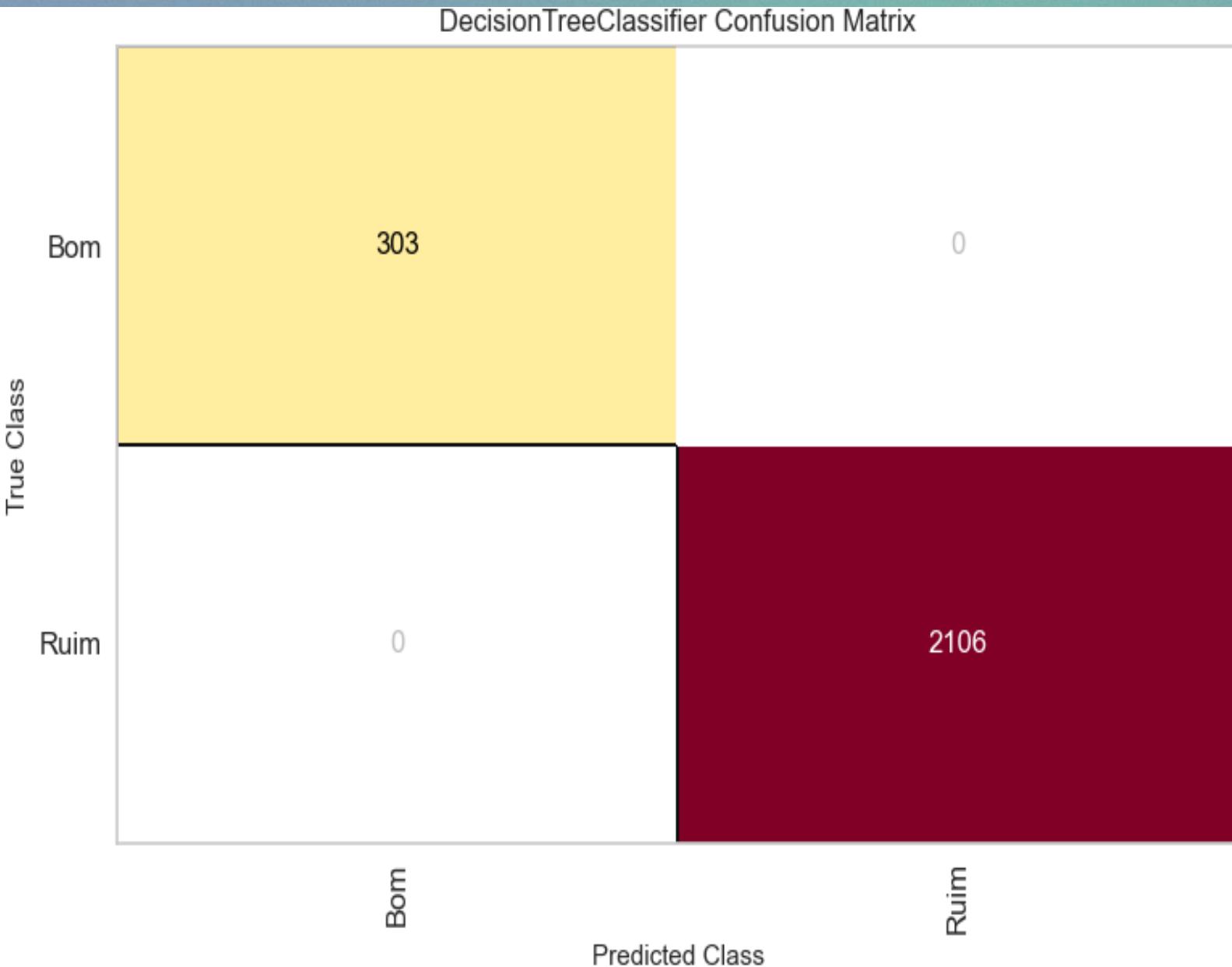
#atributos da árvore de decisão, os que tem maior valor
#representam maior ganho de informação. Importante.
print('Runtime X[0]-->',arvore.feature_importances_[0])
print('genres X[1]-->',arvore.feature_importances_[1])
print('imdb_score X[2]-->',arvore.feature_importances_[2])
print('imdb_votes X[3]-->',arvore.feature_importances_[3])
print('tmdb_score X[4]-->',arvore.feature_importances_[4])

Runtime X[0]--> 0.48137076696590986
genres X[1]--> 0.0
imdb_score X[2]--> 0.30025251796863334
imdb_votes X[3]--> 0.1775649679462195
tmdb_score X[4]--> 0.04081174711923726
```

Foi separado as variáveis previsoras da classificação, os campos categóricos convertidos em numéricos.
A árvore gerado

```
#Acurácia
print('Acurácia: %.4f' %accuracy_score(y_teste,previsoes))
Acurácia: 1.0000
```

Árvore de Decisão – Matriz de Confusão



```
# taxa de acerto - é superior ao modelo de Naive Bayes  
taxa_acerto = accuracy_score(y_teste, previsoes)  
taxa_acerto
```

1.0

```
# matriz de confusão - mostra os acertos e os erros.  
confusao = confusion_matrix(y_teste, previsoes)  
confusao
```

```
array([[ 303,    0],  
       [    0, 2106]], dtype=int64)
```

O modelo de árvore de decisão apresentou uma taxa de acerto de 100%, detectando 303 acerto no quadrante de BOM e 2106 acertos no quadrante de RUIM.

É um modelo válido para ser usado, nessa análise.

Modelos ML usados

Random Forest

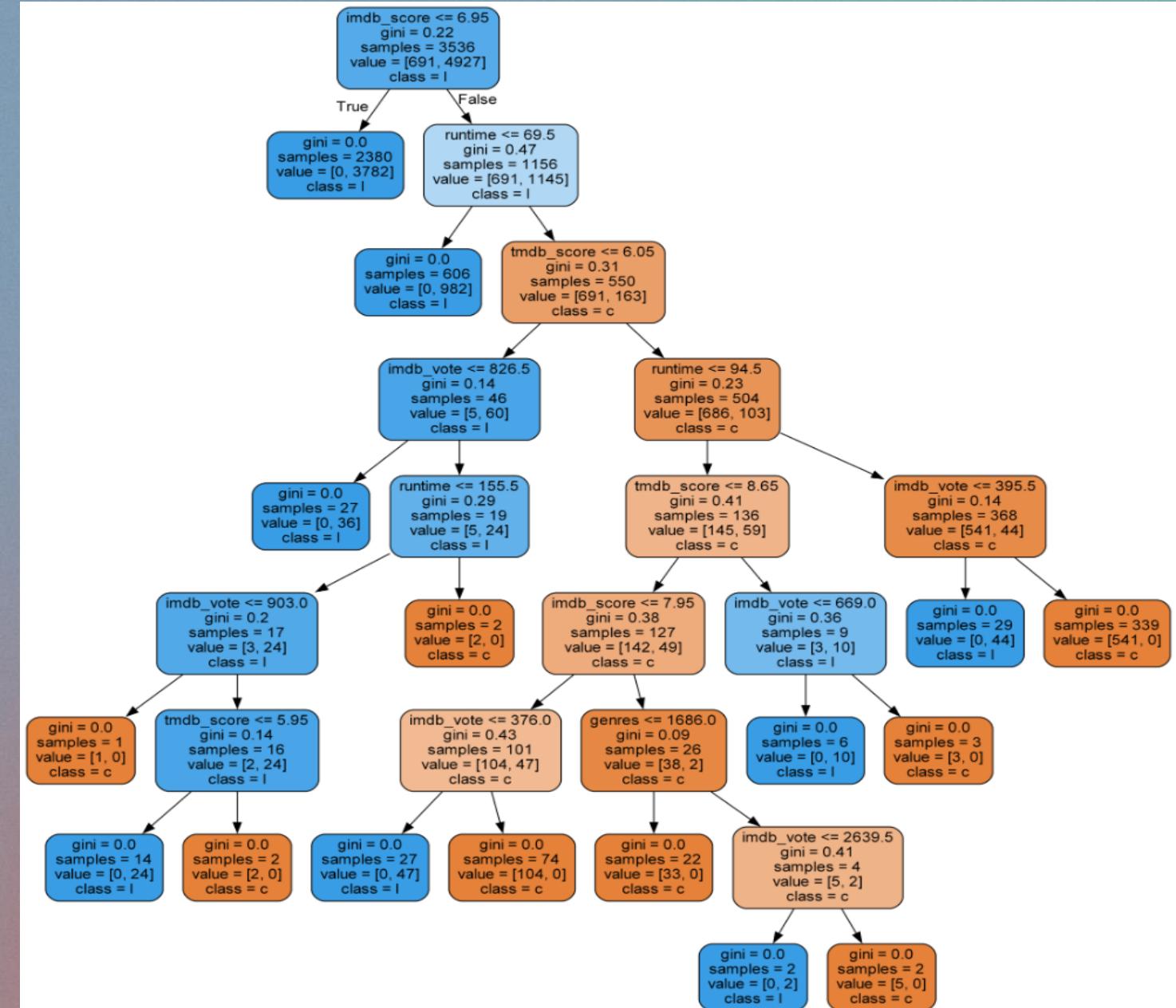
Consiste em gerar vários modelos, evitando assim o overfitting.

Cria de forma aleatória várias Árvores de Decisão e combina o resultado de todas elas para chegar no resultado final.

Muito usado no setor bancário, mercado financeiro, Hospitalar e comércio eletrônico.

- Vantagens
 - 1. Retorna de maneira muito comprehensiva a importância atribuída para cada variável independente.
 - 2. Pode ser usada para regressão e para classificação.
 - 3. Muito fácil de implementar, geralmente produz bons resultados
 - 4. Se houver árvores suficiente na floresta, o classificador não irá sobre ajustar o modelo e gerar overfitting.
- Desvantagens
 - 1. Quantidade grande de árvores pode deixar o algoritmo lento e ineficiente para previsões em tempo real.
 - 2. Para maior acurácia, precisa de mais árvores, o que faz o modelo ficar mais lento.
 - 3. Ferramenta de modelagem preditiva e não descritiva.

Random Forest- Árvore gerada

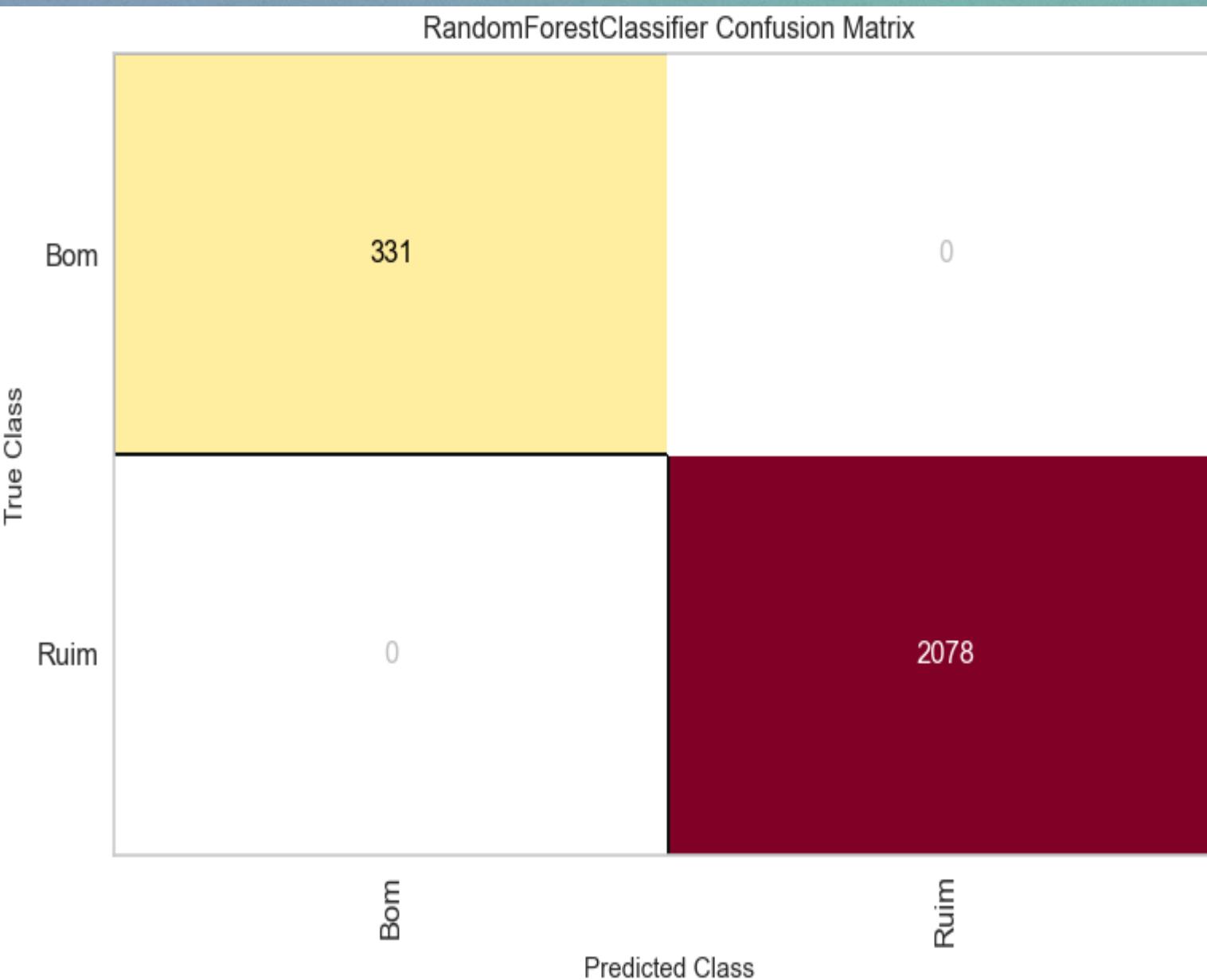


#Taxa de acerto	X[0]--> 0.41916017997266297
taxa_acerto	X[1]--> 0.010160491695535602
	X[2]--> 0.34181079540606324
	X[3]--> 0.12870301563783865
	X[4]--> 0.10016551728789942

#atributos da random forest, maiores valores tem maior ganho de informação. Mais importante.
 floresta.feature_importances_
 array([0.41916018, 0.01016049, 0.3418108 , 0.12870302, 0.10016552])

O modelo de Random foreste apresentou uma taxa de acerto de 100%, os atributos mais importantes que apresentam maior ganho de informação no modelo é runtime e imdb_score.

Análise dos dados – Matriz de Confusão



```
#Resultado da Matriz de Confusão  
confusao
```

```
array([[ 331,     0],  
       [    0, 2078]], dtype=int64)
```

```
# Acurácia  
print('Acurácia: %.4f' %accuracy_score(y_teste,previsoes))
```

Acurácia: 1.0000

O modelo de Random foreste apresentou uma acurácia de 100%, não apresentou erros e tem uma taxa no quadrante “Bom” melhor que a Árvore de Decisão.

É um modelo válido para ser usado, nessa análise.

Resultado Final

Modelos	R-Quadrática	P-Valor	Precisão	Correlação	Bom	Ruim
Regressão Linear	32%	0,09%		57%		
Naive Bayes			88%		30	2088
Árvore de Decisão			100%		303	2106
Random Forest			100%		331	2078

Os modelos de Naive Bayes, Árvore de Decisão e Random Forest, ambos apresentaram um bom desempenho nas previsões de classificações dos programas dos serviços de streaming, com destaque para Árvore de Decisão e Random Forest, que tem precisão de 100%.